

### **Amendments to the Claims:**

This listing of claims will replace all prior versions, and listing, of claims in the application:

### **Listing of the Claims:**

1. (Currently Amendment) A method of tracking references to objects of an object-oriented programming environment, said method comprising:

determining whether a command is likely to place a reference to an object on an execution stack which is used to execute computer program code in of said object-oriented programming environment;

determining whether there is a change in the flow control between the time said command is likely to place said reference to said object on said execution stack and the time said reference is used to access said object when said determining determines that said command is likely to place a reference to an object on an execution stack;

translating said command into another command when said determining determines that there is a change in the flow control; and

placing a reference to said object on a reference stack associated with said execution stack when said another command is executed, wherein said reference stack is not used to execute computer program code and is designated to store only references to objects which have been stored in a heap.

2. (Previously Presented) A method as recited in claim 1, wherein said object-oriented programming environment is a Java compliant operating environment.

3. (Original) A method as recited in claim 2, wherein said determining of whether a command is likely to place a reference on said execution stack is performed during Java Bytecode verification.

4. (Original) A method as recited in claim 3, wherein said determining of whether a command is likely to place a reference on said execution stack operates to determine whether a Getfield, Aload, Getstatic, or Return command is being performed.

5. (Canceled)

6. (Previously Presented) A method as recited in claim 4, wherein said method further comprises:

determining whether a Putfield command is likely to overwrite a reference to an object on the execution stack before said reference is used when said determining determines that there is not a change in the flow control; and

translating said command into another command when said determining determines that there is a Putfield command is likely to overwrite a reference to an object on the execution stack before said reference is used.

7. (Original) A method as recited in claim 1, wherein said reference stack and said execution stack have the same size.

8. (Original) A method as recited in claim 1, wherein at least one reference to an object is stored in an entry with an offset that is the same as the offset used to store said at least one reference in said execution stack, when said another command is executed.

9. (Currently Amended) A method of tracking references to Java objects in a Java programming environment, said method comprising:

determining whether said Java command is likely to place the only reference to a Java object on the execution stack, wherein said reference is a direct reference to said Java object which has been stored in a heap designated for storing objects;

translating said command into another command when said determining determines that said Java command is likely to place the only reference to a Java object on the execution stack;

executing said another Java command;

placing a reference to said object on a reference stack associated with said execution stack when said another command is executed; and

wherein said determining of whether said Java command is likely to place the only reference to a Java object on the execution stack is performed during Java Bytecode verification.

10. (Original) A method as recited in claim 9, wherein said determining that said Java command is likely to place the only reference to a Java object on the execution stack further comprises:

determining whether a Getfield, Aload, Getstatic, or Areturn command is being performed.

11. (Previously Presented) A method as recited in claim 10, wherein said determining that said Java command is likely to place the only reference to a Java object on the execution stack further comprises:

determining whether there is a change in the flow control.

12. (Original) A method as recited in claim 11, wherein said determining of whether said Java command is likely to place the only reference to a Java object on the execution stack further comprises:

determining whether a Putfield command is likely to overwrite a reference to an object on the execution stack before said reference is used.

13. (Original) A method as recited in claim 12, wherein said reference stack and said execution stack have the same size.

14. (Original) A method as recited in claim 13, wherein at least one reference to a Java object is stored in an entry with an offset that is the same as the offset used to store said at least one reference in said execution stack, when said another command is executed.

15. (Currently Amended) A Java Bytecode verifier suitable for operating in a Java operating environment,

wherein said Bytecode verifier operates to determine whether there is at least one Java command in a stream of Java Bytecode commands such that said at least one Java command is likely to place the only reference to a Java object on the execution stack, where said reference is a direct reference to said Java object which has been stored in a heap designated for storing objects;

wherein said Bytecode verifier operates to translate said Java command into another Java command when said Java command is likely to place the only reference to a Java object on the execution stack; and

wherein a reference associated with said command is placed on a reference stack as well as said execution stack when said another command is executed.

16. (Original) A Java Bytecode verifier as recited in claim 15, wherein said Bytecode verifier operates to determine whether a Getfield, Aload, Getstatic, or Areturn command is being performed.

17. (Original) A Java Bytecode verifier as recited in claim 16, wherein said Bytecode verifier operates to determine whether there is a change in the flow control.

18. (Original) A Java Bytecode verifier as recited in claim 16, wherein said Bytecode verifier operates to determine whether a Putfield command is likely to overwrite a reference to an object on the execution stack before said reference is used.

19. (Original) A Java Bytecode verifier as recited in claim 16, wherein said Bytecode verifier operates to:

determine whether there is a change in the flow control; and

determine whether a Putfield command is likely to overwrite a reference to an object on the execution stack before said reference is used.

20. (Currently Amended) A computer readable medium including computer program code for tracking references to objects of an object-oriented programming environment, said computer readable medium comprising:

computer program code for determining whether a command is likely to place a reference to an object on an execution stack which is used to execute computer program code in of said object-oriented programming environment;

computer program code for determining whether there is a change in the flow control between the time said command is likely to place said reference to said object on said execution stack and the time said reference is used to access said object when said determining determines that said command is likely to place a reference to an object on an execution stack;

computer program code for translating said command into another command when said determining determines that there is a change in the flow control; and

computer program code for placing a reference to said object on a reference stack associated with said execution stack when said another command is executed, wherein said reference stack is not used to execute computer program code and is designated to store only references to objects which have been stored in a heap.

21. (Previously Presented) A computer readable medium as recited in claim 20, wherein said object-oriented programming environment is a Java compliant operating environment.